

【ES6】 async / await

使用方式

```
// #1 當資料有順序性的時候
const fn1 = () => {
  axios.get("https://randomuser.me/api/")
    .then((res) => {
      console.log("fn1 seed", res.data.info.seed);
      const seed = res.data.info.seed;
      return axios.get(`https://randomuser.me/api/?seed=${seed}`);
    })
    .then((res) => {
      console.log("fn1 seed2", res.data.info.seed);
    })
};

// #2 使用 async await 來修改
const fn2 = async () => {
  const res = await axios.get("https://randomuser.me/api/")
  const seed = res.data.info.seed;
  console.log("fn2 seed", res.data.info.seed);
  const res2 = await axios.get(`https://randomuser.me/api/?seed=${seed}`);
  const seed2 = res2.data.info.seed;
  console.log("fn2 seed2", res2.data.info.seed);
};

// #3 立即函式的寫法
(async function(){
  const res = await axios.get("https://randomuser.me/api/")
  const seed = res.data.info.seed;
  console.log("fn3 seed", res.data.info.seed);
  const res2 = await axios.get(`https://randomuser.me/api/?seed=${seed}`);
  const seed2 = res2.data.info.seed;
  console.log("fn3 seed2", res2.data.info.seed);
})();
```

錯誤處理

```
const fn4 = async () => {
  try {
    const res = await axios.get("https://randomuser.me/api/")
    return res
  } catch (error) {
    throw error
  }
};

fn4()
  .then((res) => {
    console.log(res);
  })
  .catch((err) => {
    console.log(err);
  });
```

async function 本身就是 promise

```

<script src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"></script>
<script type="text/babel">
// #1 Async function 的本質
// 1. async function 本身就是 Promise
// 2. await 運算子，它的後方需要加入的是 Promise

// const asyncFn = async () => {
//   const res = await axios.get('https://jsonplaceholder.typicode.com/todos/1');
// }

// console.log(asyncFn());

// #2 Async function 回傳值
const asyncFn = async () => {
  const res = await axios.get('https://jsonplaceholder.typicode.com/todos/1');
  // console.log(res);
  res.data.name = '卡斯伯'
  return res;
}
asyncFn()
  .then(res => {
    console.log(res);
  })

```

You, 4 秒前 • Uncommitted changes

async Function 與 React

```

// async Function 與 React
const App = () => {
  const [data, setData] = React.useState({})

  React.useEffect(() => {
    (async () => {
      const res = await axios.get('https://jsonplaceholder.typicode.com/todos/1');
      console.log(res);
      setData(res.data);
    })();
  }, []);

  return <div>
    {data.title}
  </div>
}

```

🕒 修訂版本 #4

★ 由 treeman 建立於 27 🕒 2023 07:32:38

🔧 由 treeman 更新於 5 🕒 2023 10:14:59